

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-297642

(43)Date of publication of application : 12.11.1996

(51)Int.Cl.

G06F 15/16
G06F 15/163

(21)Application number : 07-102532

(71)Applicant : KOFU NIPPON DENKI KK

(22)Date of filing : 26.04.1995

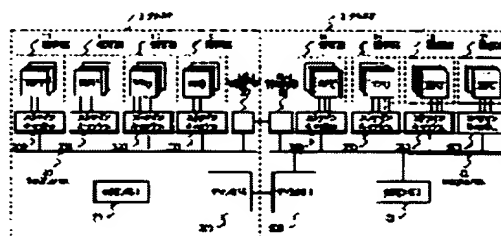
(72)Inventor : HORIKAWA KOICHI

(54) SHARED MEMORY TYPE MULTIPROCESSOR SYSTEM

(57)Abstract:

PURPOSE: To improve performance by improving the hit rate of cache by guaranteeing coherency between caches concerning the directory system shared memory type multiprocessor system provided with plural store-in caches.

CONSTITUTION: Directory control parts 100 and 200 monitor requests on system buses 40 and 41 of their own clusters and transfer requests or data to the directory control part of the other cluster as needed. When the address competition discrimination circuit detects the coincidence of addresses between the request on the system bus of the present cluster and the request from the directory control part of the other cluster, 'cancel' is issued onto the system bus of the present cluster and the request issued onto the system bus of the present cluster is tried again. Besides, the system buses 40 and 41 are controlled while shifting their cycles by half with each other.



LEGAL STATUS

[Date of request for examination] 26.04.1995

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2786124

[Date of registration] 29.05.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-297642

(43) 公開日 平成8年(1996)11月12日

(51) Int.Cl.⁶

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/16
15/163

G 0 6 F 15/16

4 0 0 M
3 2 0 K

審査請求 有 請求項の数 2 O L (全 10 頁)

(21) 出願番号 特願平7-102532

(22) 出願日 平成7年(1995)4月26日

(71) 出願人 000168285

甲府日本電気株式会社

山梨県甲府市大津町1088-3

(72) 発明者 堀川 浩一

山梨県甲府市大津町1088-3 甲府日本電
気株式会社内

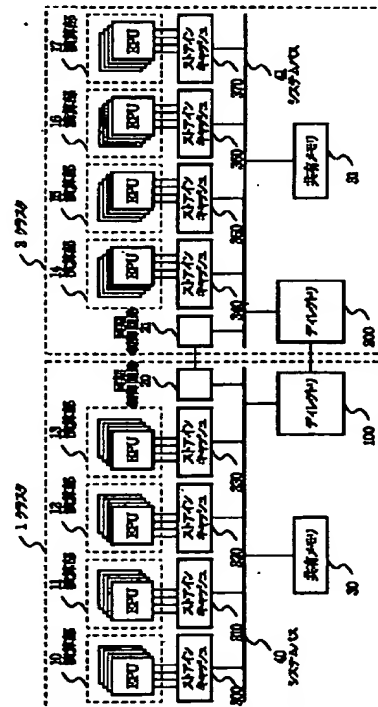
(74) 代理人 弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 共有メモリ型マルチプロセッサシステム

(57) 【要約】

【目的】 複数のストアインキャッシュを持つディレ
クトリ方式の共有メモリ型マルチプロセッサシステムに
おいて、キャッシュ間のコヒーレンスを保証し、キャッ
シュのヒット率を向上させ性能向上を図ることを目的と
する。

【構成】 ディレクトリ制御部100および200が
自クラスタのシステムバス40および41上のリクエ
ストを監視し、必要に応じて他クラスタのディレクトリ制
御部にリクエストやデータを転送する。またアドレス競
合判定回路102が自クラスタのシステムバス上のリク
エストと他クラスタのディレクトリ制御部からのリク
エストのアドレス一致を検出した場合、自クラスタのシ
ステムバス上に「キャンセル」を発行し、自クラスタのシ
ステムバス上に発行されたリクエストを再試行させる。
また、システムバス40および41は互いに半周期ずら
して制御される。



【特許請求の範囲】

【請求項 1】 2つのクラスタを有し、各クラスタが、複数のプロセッサと前記プロセッサの全てに接続された1つのストアインキャッシュとを有する複数の処理装置と、共有メモリと、前記複数の処理装置と前記共有メモリとを接続するシステムバスとを備えた共有メモリ型マルチプロセッサにおいて、前記クラスタの各々が、

(a) 前記システムバス間の同期を制御する同期制御手段と、(b) 自クラスタの前記システムバスと接続されかつ他クラスタのディレクトリと相互に接続され他クラスタのストアインキャッシュに保持された自クラスタ内共有メモリのメモリブロックアドレスを貯蔵するディレクトリと、(c) 自クラスタの前記処理装置から前記共有メモリへのメモリアクセス要求のアドレスと他クラスタの前記ディレクトリからのアドレスとを受け取りそれらのアドレスの一致を検出するアドレス一致検出手段と、(d) 前記アドレス一致検出手段の結果に応じて自クラスタの前記メモリアクセス要求をキャンセルするキャンセル信号を発行するキャンセル手段と、(e) 前記ストアインキャッシュに設けられキャンセル信号を受け取り自クラスタの前記メモリアクセス要求を無効化し再試行する再試行手段と、を具備したことを特徴とする共有メモリ型マルチプロセッサシステム。

【請求項 2】 前記ディレクトリ内のアドレスの各々に対応して前記ディレクトリ内に更新フラグを設け、他クラスタのストアインキャッシュに保持された自クラスタの共有メモリのメモリブロックが他クラスタのプロセッサによる書き込みで更新されている場合に、自クラスタのディレクトリに貯蔵されている前記メモリブロックアドレスに対応する前記更新フラグをセットするようにしたことを特徴とする請求の範囲の第 1 項に記載の共有メモリ型マルチプロセッサシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は共有メモリ型マルチプロセッサシステムに関し、特にディレクトリ方式の共有メモリ型マルチプロセッサシステムに関する。

【0002】

【従来の技術】 1つのシステムバスに、複数のストアインキャッシュおよび共有メモリが接続されているマルチプロセッサシステム（一般的に「クラスタ」と呼ばれる）を、ディレクトリ方式で複数接続した共有メモリ型マルチプロセッサシステムは既知の技術で実現されている。たとえば特開平 2-12361 号公報記載の「階層化バスによる並列計算機システム」、特開平 4-291446 号公報記載の「スケーラブルメモリ帯域を備えた緊密結合型多重プロセッサ」がある。しかし、特開平 2-12361 号公報の技術では、クラスタ間のストアインキャッシュのコヒーレンシの保証の方法が開示されていない。また、特開平 4-291446 号公報の技術で

はクラスタ相互を接続するのにクロスバ（これに接続されている複数のクラスタのうち、任意の 2 つのクラスタ間の通信を可能にするスイッチ）を用いており、さらにストアインキャッシュのコヒーレンシを保証するため、クラスタ内、クラスタ間を問わず複数のストアインキャッシュ間で同一メモリブロックを同時にキャッシング（キャッシュに貯蔵することをキャッシングするという。以下この用語を使用する。）しないように制御している。

10 【0003】

【発明が解決しようとする課題】 上述した特開平 2-12361 号公報記載の構成では、クラスタ間のストアインキャッシュのコヒーレンシの保証がとれないという欠点がある。また、特開平 4-291446 号公報記載の構成ではクラスタ単体だけでも動作することができるシステムを 2 つ接続した構成とする場合において、バス方式でなくクロスバ方式を採用しているために、ハードウェア量の増大を招くという欠点がある。さらに、ストアインキャッシュのコヒーレンシの保証のために、共有メモリのいずれのメモリブロックも同時期にはどれか 1 個のストアインキャッシュにしかキャッシングできないように制御される。このため、ストアインキャッシュでのヒット率が低下しシステム全体の性能が低下するという欠点がある。

20 【0004】

【課題を解決するための手段】 本発明の第 1 の共有メモリ型マルチプロセッサシステムは、2 つのクラスタを有し、各クラスタが複数のプロセッサと前記プロセッサの全てに接続されたストアインキャッシュとを有する複数の処理装置と、共有メモリと、前記複数の処理装置と前記共有メモリとを接続するシステムバスとを備えた共有メモリ型マルチプロセッサであり、前記クラスタの各々が、(a) 前記システムバス間の同期を制御する同期制御手段と、(b) 自クラスタの前記システムバスと接続されかつ他クラスタのディレクトリと相互に接続され他クラスタのストアインキャッシュに保持された自クラスタ内共有メモリのメモリブロックアドレスを貯蔵するディレクトリと、(c) 自クラスタの前記処理装置から前記共有メモリへのメモリアクセス要求のアドレスと他クラスタの前記ディレクトリからのアドレスとを受け取りそれらのアドレスの一致を検出するアドレス一致検出手段と、(d) 前記アドレス一致検出手段の結果に応じて自クラスタの前記メモリアクセス要求をキャンセルするキャンセル信号を発行するキャンセル手段と、(e) 前記ストアインキャッシュに設けられキャンセル信号を受け取り自クラスタの前記メモリアクセス要求を無効化し再試行する再試行手段と、を備える。

50 【0005】 本発明の第 2 の共有メモリ型マルチプロセッサシステムは、上記第 1 の構成に加え、前記ディレクトリ内のアドレスの各々に対応して前記ディレクトリ内

3

に更新フラグを設け、他クラスタのストアインキャッシュに保持された自クラスタの共有メモリのメモリブロックが他クラスタのプロセッサによる書き込みで更新されている場合に、自クラスタのディレクトリに貯蔵されている前記メモリブロックアドレスに対応する前記更新フラグをセットする手段を備える。

【0006】

【実施例】本発明について、図面を参照して詳細に説明する。

【0007】図1は本発明の一実施例を示すブロック図である。

4

【0008】図1を参照すると、演算部10～17はそれぞれ4つのプロセッサ（以下EPUという）からなり、それぞれストアインキャッシュ300～370に接続されている。また、ストアインキャッシュ300～330はシステムバス40に接続され、ストアインキャッシュ340～370はシステムバス41に接続されている。本実施例に用いたストアインキャッシュにキャッシングされるメモリブロックのとりうる状態は表1の通りである。

【0009】

【表1】

状態	意味
IV	キャッシングしていない。
CE	システム内で唯一キャッシングしている。但し更新していない。
CS	キャッシングしている。システム内の他のストアインキャッシュが同じメモリブロックをキャッシングしている可能性がある。ただし更新していない。
DE	システム内で唯一キャッシングしている。ただし更新している。

【0010】システムバス40および41は、本実施例の場合、ともに5クロックサイクル（5サイクル）で1周期となるよう制御される（表2）。システムバス40および41は、同期制御回路20および21にそれぞれ接続され、同期制御回路20および21は互いに接続される。システムバス40および41は互いに同期をずら

して制御される。本実施例の場合、2.5サイクルずらしている。すなわち、システムバス40のステート1が、システムバス41のステート3と4の間にくることになる。

【0011】

【表2】

ステート	1	2	3	4	5
状態	RQ	AR0	AD0	AD1	AR1
RQ	システムバス獲得要求、および、自クラスタリクエストキャンセルを発行する。				
AR0	自クラスタ内調停。自クラスタのどのストアインキャッシュがシステムバスにリクエストを発行できるかを定める。システムバスには何も発行しない。				
AD0	アドレス上位およびリクエストコマンドを発行する				
AD1	アドレス下位を発行する。				
AR1	自、他クラスタアドレス競合判定を行う。アドレス競合が起こったら次のクロックサイクルすなわちステート1で自クラスタリクエストキャンセルを発行する。				

【0012】共有メモリ30はシステムバス40に接続され、システムバス40上に発行されるリクエストに対して、必要に応じてデータを返す。共有メモリ31も同

様であり、共有メモリ30、31はシステム内の全てのEPUからアクセスされる可能性がある。ディレクトリ100は、システムバス40およびディレクトリ200

5

に接続され、またディレクトリ200はシステムバス41に接続される。

【0013】図2は図1の実施例に使用されるストアインキャッシュのブロック図である。ここではストアインキャッシュ300について示してあるが、他のストアインキャッシュも同様の構成である。

【0014】図2を参照すると、ストアインキャッシュ制御回路302は演算部10、ストアインキャッシュメモリ301、再試行指示回路303、およびシステムバス40に接続されている。

【0015】ストアインキャッシュ制御回路302は、接続されているEPUからの要求に応じて、ストアインキャッシュメモリ301からEPUにデータを返したり、ストアインキャッシュメモリ301を更新したり、システムバス40にメモリブロックリード等のリクエスト（以下単にリクエストという）を発行する。

【0016】また、再試行指示回路303はシステムバス40上に「キャンセル」が発行された場合、ストアインキャッシュ制御回路302に対し、リクエストを再試行するよう指示する。

【0017】図3は図1の実施例に使用されるディレクトリのブロック図である。ここではディレクトリ100について示してあるが、ディレクトリ200も同様の構成である。図3を参照するとディレクトリ制御回路102はシステムバス40、ディレクトリメモリ101、およびディレクトリ200と接続される。ディレクトリメモリ101には、クラスタ2のいずれかのストアインキャッシュがクラスタ1の共有メモリ30のメモリブロックをキャッシングしている場合、そのメモリブロックのアドレスが登録される。またそのメモリブロックがクラスタ2のいずれかのストアインキャッシュ内で更新されているかどうかの更新フラグも同時に保持される。

【0018】ディレクトリ制御回路102は、システムバス40上に発行されたリクエストを監視し、ディレクトリメモリ101内の情報を索引、更新し、必要に応じてディレクトリ200にリクエストを転送する。また逆に、ディレクトリ200から転送されるリクエストに応じてディレクトリ101内の情報を索引、更新し、必要に応じてシステムバス40にリクエストを発行する（ディレクトリ100とディレクトリ200の間の転送は1.5サイクルかかるものとする）。ディレクトリ制御回路102がこのようにディレクトリメモリ101の情報を登録、索引、更新することを、単に「ディレクトリに登録、を索引、を更新」という。

【0019】アドレス一致検出回路103はシステムバス40およびディレクトリ200からのリクエストのアドレス一致を検出し、一致が検出されるとキャンセル信号生成回路104に通知する。

【0020】キャンセル信号生成回路104は、アドレス一致検出回路103からの通知により、システムバス

6

40上に「キャンセル」を発行する。

【0021】図4は、図3のディレクトリメモリ101のブロック図である。図4を参照すると、ディレクトリメモリ101の各エントリ0からNのアドレスは、クラスタ2のいずれかのストアインキャッシュにキャッシングされている共有メモリ30のメモリブロックのアドレスである。更新フラグは対応するアドレスのメモリブロックがクラスタ2のストアインキャッシュ内で更新されると「1」にセットされる。有効フラグは各エントリが有効かどうかを示すものであり、「1」にセットされていなければそのエントリのアドレスは、無効であり、ディレクトリメモリ101には登録されていないとみなされる。

【0022】次に、図1、図2、図3、図4のブロック図を参照して本発明の実施例の動作について詳細に説明する。便宜的にクラスタ1の演算部10のいずれかのEPUから共有メモリ30または31をアクセスしようとした場合について説明する。クラスタ1の他のEPUおよびクラスタ2のEPUの場合も全く同様である。EPUからの要求の種類に応じて、以下の4つの場合の基本動作に分けて説明する。

1. 自クラスタのメモリ空間（共有メモリ30）に対するリード（読み出し）
2. 他クラスタのメモリ空間（共有メモリ31）に対するリード（読み出し）
3. 自クラスタのメモリ空間（共有メモリ30）に対するライト（書き込み）
4. 他クラスタのメモリ空間（共有メモリ31）に対するライト（書き込み）

1. 自クラスタのメモリ空間（共有メモリ30）に対するリードの場合を説明する。以下、1. 1. 所望のデータがストアインキャッシュ300内にキャッシングされている場合と、1. 2. 所望のデータがストアインキャッシュ300内にキャッシングされていない場合とに分けて説明する。

【0023】1. 1. 所望のデータがストアインキャッシュ300内にキャッシングされている場合、そのデータがEPUに返される。

【0024】1. 2. 所望のデータがストアインキャッシュ300内にキャッシングされていない場合、ストアインキャッシュ制御回路302はシステムバス40のステート1でシステムバス獲得要求を発行し、システムバス40が獲得できたら、システムバス40のステート3、4で「共有ブロックリードリクエスト」（以下SBRという）を発行する（以降、ストアインキャッシュがシステムバスにリクエストを発行する時は同様の動作をするものとする）。このSBRに対するデータリプライが返って来たらこれをEPUに返すことになる。以下、1. 2. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしていた場合と、1. 2. 2.

7

クラスタ1の他のストアインキャッシュがCEまたはCS状態でキャッシングしていた場合と、1. 2. 3. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合とに分けて説明する。

【0025】1. 2. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしていた場合、そのストアインキャッシュがシステムバス40上にデータリブライを行ない、自身をCS状態にする。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。共有メモリ30もこのデータを受けとり、メモリの内容を更新する。

【0026】1. 2. 2. クラスタ1の他のストアインキャッシュがCEまたはCS状態でキャッシングしていた場合、CE状態の場合はCS状態に更新する。共有メモリ30がシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、CS状態で登録し、EPUに返す。

【0027】1. 2. 3. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合、ディレクトリ100は、ディレクトリメモリ101を索引する。以下、1. 2. 3. 1. アドレスがディレクトリ100に登録されていない場合と、1. 2. 3. 2. アドレスがディレクトリ100に登録されておりかつ更新フラグが‘0’である場合と、1. 2. 3. 3. アドレスがディレクトリ100に登録されておりかつ更新フラグが‘1’である場合とに分けて説明する。

【0028】1. 2. 3. 1. アドレスがディレクトリ100に登録されていない場合は何もしない。この時は共有メモリ30がシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、CE状態で登録し、EPUに返す。

【0029】1. 2. 3. 2. アドレスがディレクトリ100に登録されておりかつ更新フラグが‘0’である場合は何もしない。この時は共有メモリ30がデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。

【0030】1. 2. 3. 3. アドレスがディレクトリ100に登録されておりかつ更新フラグが‘1’である場合、更新フラグを‘0’にする。また、ディレクトリ200にSBRを転送する。ディレクトリ200はシステムバス41上にSBRを発行する。このSBRに対し、クラスタ2のいずれかのストアインキャッシュのうち、DE状態でキャッシングしているものがシステムバス41上にデータリブライを行なう。データリブライを行なったストアインキャッシュは状態をDEからCSに更新する。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、CS状態で登録し、EPUに返す。共有メモリ30もこ

8

のデータを受けとり、メモリの内容を更新する。

【0031】2. 他クラスタのメモリ空間（共有メモリ31）に対するリードの場合を説明する。以下、2.

1. 所望のデータがストアインキャッシュ300内にキャッシングされている場合と、2. 2. 所望のデータがストアインキャッシュ300内にキャッシングされていない場合とに分けて説明する。

【0032】2. 1. 所望のデータがストアインキャッシュ300内にキャッシングされていればそのデータがEPUに返される。

【0033】2. 2. 所望のデータがストアインキャッシュ300内にキャッシングされていなければ、システムバス40にSBRを発行する。このSBRに対するデータリブライが返って来たらこれをEPUに返すことになる。以下、2. 2. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしていた場合と、2. 2. 2. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしていた場合と、2. 2. 3. クラスタ1の他のストアインキャッシュのどれもがキャッシングしていない場合とに分けて説明する。

【0034】2. 2. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしていた場合、そのストアインキャッシュがシステムバス40上にデータリブライを行ない、自身をCS状態にする。ストアインキャッシュ300はこのデータを受けとり、CS状態で登録し、EPUに返す。これと並行してディレクトリ100がデータを受けとり、ディレクトリ200に転送する。ディレクトリ200はこのデータのメモリブロックアドレスに対する更新フラグを‘0’にする。また、システムバス41上に「メモリブロックライトリクエスト」を発行し、共有メモリ31にこのデータをライトする。

【0035】2. 2. 2. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしていた場合、この（これらの）ストアインキャッシュは何もしない。ディレクトリ100はSBRをディレクトリ200に転送する。ディレクトリ200はこのSBRをシステムバス41上に発行する。共有メモリ31がシステムバス41上にデータリブライを行なう。ディレクトリ200はこのデータを受けとり、ディレクトリ100に転送する。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。

【0036】2. 2. 3. クラスタ1の他のストアインキャッシュのどれもがキャッシングしていない場合、ディレクトリ100はSBRをディレクトリ200に転送する。ディレクトリ200はこのSBRのアドレスを、更新フラグを‘0’、有効フラグを‘1’として登録する。また、システムバス41上にSBRを発行する。以下、このSBRに対する動作を2. 2. 3. 1. クラス

タ2のストアインキャッシュいずれかがDE状態でキャッシングしていた場合と、2. 2. 3. 2. クラスタ2のストアインキャッシュがCEまたはCS状態でキャッシングしていた場合と、2. 2. 3. 3. クラスタ2のストアインキャッシュどれもがキャッシングしていない場合とに分けて説明する。

【0037】2. 2. 3. 1. クラスタ2のストアインキャッシュいずれかがDE状態でキャッシングしていた場合、そのストアインキャッシュがシステムバス41上にデータリブライを行ない、自身をCS状態にする。ディレクトリ200はこのデータを受けとり、ディレクトリ100に転送する。共有メモリ31もこのデータを受けとり、メモリの内容を更新する。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。

【0038】2. 2. 3. 2. クラスタ2のストアインキャッシュがCEまたはCS状態でキャッシングしていた場合、CE状態の場合はCS状態に更新する。共有メモリ31がシステムバス41上にデータリブライを行なう。ディレクトリ200はこのデータを受けとり、ディレクトリ100に転送する。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。

【0039】2. 2. 3. 3. クラスタ2のストアインキャッシュどれもがキャッシングしていない場合、共有メモリ31がシステムバス41上にデータリブライを行なう。ディレクトリ200はこのデータを受けとり、ディレクトリ100に転送する。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、CS状態で登録し、EPUに返す。

【0040】3. 自クラスタのメモリ空間（共有メモリ30）に対するライトの場合を説明する。以下、3. 1. ストアインキャッシュ300が、ライト対象のデータをDE状態でキャッシングしている場合と、3. 2. ストアインキャッシュ300が、ライト対象のデータをCE状態でキャッシングしている場合と、3. 3. ストアインキャッシュ300が、ライト対象のデータをCS状態でキャッシングしている場合と、3. 4. ストアインキャッシュ300が、ライト対象のデータをキャッシングしていない場合とに分けて説明する。

【0041】3. 1. ストアインキャッシュ300が、ライト対象のデータをDE状態でキャッシングしている場合、これにライトする。

【0042】3. 2. ストアインキャッシュ300が、ライト対象のデータをCE状態でキャッシングしている場合、これにライトし、自身をDE状態にする。

【0043】3. 3. ストアインキャッシュ300が、

ライト対象のデータをCS状態でキャッシングしている場合、これにライトし、DE状態にする。これと並行してシステムバス40上に「メモリブロック無効化リクエスト」（以下INVという）を発行する。クラスタ1の他のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にして登録を抹消する。ディレクトリ100に更新フラグが‘0’で登録されている場合には、ディレクトリ100は有効フラグを‘0’として登録を抹消し、ディレクトリ200にINVを転送する。ディレクトリ200はINVをシステムバス41上に発行する。クラスタ2のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。

【0044】3. 4. ストアインキャッシュ300がライト対象のデータをキャッシングしていない場合、システムバス40上に「排他ブロックリードリクエスト」

（以下EBRという）を発行する。以下、3. 4. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしている場合と、3. 4. 2. クラスタ1の他のストアインキャッシュがCE状態でキャッシングしている場合と、3. 4. 3. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしている場合と、3. 4. 4. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合とに分けて説明する。

【0045】3. 4. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしている場合、そのストアインキャッシュがシステムバス40上にデータリブライを行ない、自身をIV状態にする。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0046】3. 4. 2. クラスタ1の他のストアインキャッシュがCE状態でキャッシングしている場合、これをIV状態にする。共有メモリ30がシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、これに対してライトし、DE状態で登録する。

【0047】3. 4. 3. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしている場合、これをIV状態にする。共有メモリ30がシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はデータを受けとり、これに対してライトし、DE状態で登録する。また、これらと並行してディレクトリ100はディレクトリを索引し、更新フラグが‘0’で登録されているならば有効フラグ‘0’として登録を抹消し、ディレクトリ200にINVを転送する。ディレクトリ200はシステムバス41上にINVを発行する。クラスタ2のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。

【0048】3. 4. 4. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合、ディレクトリ100は、ディレクトリメモリ101を索引する。以下、3. 4. 4. 1. ディレクトリ100に登録されていない場合と、3. 4. 4. 2. ディレクトリ100に更新フラグが‘0’で登録されている場合と、3. 4. 4. 3. ディレクトリ100に更新フラグが‘1’で登録されている場合とに分けて説明する。

【0049】3. 4. 4. 1ディレクトリ100に登録されていない場合、何もしない。この場合、共有メモリ30がシステムバス40上にデータリプライを行なう。ストアインキャッシュ300はデータを受けとり、これに対してライトし、DE状態で登録する。

【0050】3. 4. 4. 2. ディレクトリ100に更新フラグが‘0’で登録されている場合、ディレクトリ100は有効フラグを‘0’として登録を抹消し、ディレクトリ200にINVを転送する。ディレクトリ200はシステムバス41上にINVを発行する。クラスタ2のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。これらと並行して、共有メモリ30がシステムバス40上にデータリプライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0051】3. 4. 4. 3. ディレクトリ100に更新フラグが‘1’で登録されている場合、有効フラグを‘0’として登録を抹消し、ディレクトリ200にEBRを転送する。ディレクトリ200はシステムバス41上にEBRを発行する。このEBRに対し、クラスタ2のいずれかのストアインキャッシュのうち、DE状態でキャッシングしているものがシステムバス41上にデータリプライを行なう。データリプライを行なったストアインキャッシュは状態をDEからIVに更新する。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリプライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0052】4. 他クラスタのメモリ空間（共有メモリ31）に対するライトの場合を説明する。以下、4. 1. ストアインキャッシュ300が、ライト対象のデータをDE状態でキャッシングしている場合と、4. 2. ストアインキャッシュ300が、ライト対象のデータをCS状態でキャッシングしている場合と、4. 3. ストアインキャッシュ300がライト対象のデータをキャッシングしていない場合とに分けて説明する。

【0053】4. 1. ストアインキャッシュ300が、ライト対象のデータをDE状態でキャッシングしている場合、これにライトする。

【0054】4. 2. ストアインキャッシュ300が、

ライト対象のデータをCS状態でキャッシングしている場合、これにライトし、DE状態にする。これと並行してシステムバス40上にINVを発行する。クラスタ1の他のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。ディレクトリ100はディレクトリ200にINVを転送する。ディレクトリ200は更新フラグを‘1’にし、INVをシステムバス41上に発行する。クラスタ2のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。

【0055】4. 3. ストアインキャッシュ300がライト対象のデータをキャッシングしていない場合、システムバス40上にEBRを発行する。以下、4. 3. 1. クラスタ1の他のストアインキャッシュがDE状態でキャッシングしている場合、4. 3. 2. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしている場合、4. 3. 3. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合とに分けて説明する。

【0056】4. 3. 1クラスタ1の他のストアインキャッシュがDE状態でキャッシングしている場合、そのストアインキャッシュがシステムバス40上にデータリプライを行ない、自身をIV状態にする。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0057】4. 3. 2. クラスタ1の他のストアインキャッシュがCS状態でキャッシングしている場合、そのストアインキャッシュは、これをIV状態にする。ディレクトリ100は、ディレクトリ200にEBRを転送する。ディレクトリ200は更新フラグを‘1’にして、システムバス41上にEBRを発行する。クラスタ2のストアインキャッシュでCS状態でキャッシングしているものがあれば、これをIV状態にする。共有メモリ31がシステムバス41上にデータリプライを行なう。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリプライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0058】4. 3. 3. クラスタ1の他のストアインキャッシュどれもがキャッシングしていない場合、ディレクトリ100は、ディレクトリ200にEBRを転送する。ディレクトリ200は、更新フラグを‘1’、有効フラグを‘1’として登録し、システムバス41上にEBRを発行する。以下、このEBRに対する動作を4. 3. 3. 1. クラスタ2のストアインキャッシュいずれかがDE状態でキャッシングしている場合と、4. 3. 3. 2. クラスタ2のストアインキャッシュがCE状態またはCS状態でキャッシングしている場合と、

4. 3. 3. 3. クラスタ2のストアインキャッシュど

れもがキャッシングしていない場合とに分けて説明する。

【0059】4. 3. 3. 1. クラスタ2のストアインキャッシュいずれかがDE状態でキャッシングしている場合、そのストアインキャッシュがシステムバス41上にデータリブライを行ない、自身をIV状態にする。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0060】4. 3. 3. 2. クラスタ2のストアインキャッシュがCE状態またはCS状態でキャッシングしている場合、これをIV状態にする。共有メモリ31がシステムバス41上にデータリブライを行なう。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0061】4. 3. 3. 3. クラスタ2のストアインキャッシュどれもがキャッシングしていない場合、共有メモリ31がシステムバス41上にデータリブライを行なう。このデータはディレクトリ200で受けとられ、ディレクトリ100に転送される。ディレクトリ100はシステムバス40上にデータリブライを行なう。ストアインキャッシュ300はこのデータを受けとり、これに対してライトし、DE状態で登録する。

【0062】次にアドレス一致検出と、キャンセルとの動作を図1、図2、図3を参照して詳細に説明する。

【0063】システムバス40にリクエストが発行された時（システムバス40のステート3、4のタイミング）、ディレクトリ200からリクエストが転送されて来なければ何もしない。

【0064】ディレクトリ200がディレクトリ100にリクエストを転送する時には、システムバス41のステート5、1のタイミングで行なう。このリクエストはディレクトリ100によりシステムバス40のステート4、5のタイミングで受けとられる。これと並行して、ディレクトリ100はシステムバス40上のリクエストをシステムバス40のステート4、5のタイミングで受けとる。したがって、アドレス一致検出回路103は、システムバス40からのリクエストのアドレスと、ディレクトリ200からのリクエストのアドレスを、システムバス40のステート5のタイミングで比較することができる。もし一致している時にはキャンセル信号生成回路104に通知する。キャンセル信号生成回路104はアドレス一致検出回路103からの指示により、システムバス40上の「キャンセル」を発行する（システムバス40の次の周期のステート1のタイミング）。

【0065】システムバス40にリクエストを発行したストアインキャッシュが、たとえばストアインキャッシュ300だったとする（図2参照）と、システムバス40上の「キャンセル」は再試行指示回路303により検出され、再試行指示回路303はストアインキャッシュ制御回路302に、そのリクエストを再試行するよう指示する。ストアインキャッシュ制御回路302は、内部でそのリクエストを無効化し、再度同じリクエストをシステムバスの次の周期以降で発行する。以上、システムバス40上に一度発行されたリクエストがキャンセルされる場合を説明したが、ディレクトリ200内でアドレス一致が検出された場合には、システムバス41上のリクエストが同様にキャンセルされることになる。

【0066】システムバス40および41は互いに2.5サイクルずらして制御されているので、信号線105と106上をリクエストが同時に通過することはない。アドレス一致検出回路103（またはクラスタ2のアドレス一致検出回路）で必ずアドレス一致の検出が可能である。すなわち、アドレス一致検出のタイミングおよび場所は、「システムバス40のステート5のタイミングでディレクトリ100内」→「システムバス41のステート5のタイミングでディレクトリ200内」→「システムバス40のステート5のタイミングでディレクトリ100内」・・・と、クラスタ1と2で交互に行なわれることになる。したがって、2つのクラスタからのリクエストのアドレスを保持しておく等特別なハードウェア機構を必要としない。また、システムバス40上に発行されたリクエストでクラスタ2に転送されるものは、アドレス一致検出回路103でアドレス一致が検出されない限り待たされずにクラスタ2に転送され、クラスタ2のシステムバス41が獲得できしだい発行できる。また、システムバス41上に発行されたリクエストでクラスタ1に転送されるものについても同様である。

【0067】なお、EPUの台数、ストアインキャッシュの個数等本実施例以外の構成でも本発明の原理を用いれば、同様に実現できることはいうまでもない。

【0068】

【発明の効果】以上説明したように、本発明による共有メモリ型マルチプロセッサシステムは、クラスタ間のストアインキャッシュのコヒーレンシの保証ができるという効果がある。また、2つのクラスタを接続するのにクロスバを用いない構成とすることによりハードウェア量を少なくできるという効果がある。またさらに、クラスタ内、クラスタ間を問わず複数のストアインキャッシュ間で同一メモリブロックを共有できるように制御することにより、ストアインキャッシュのヒット率が向上しシステム全体の性能向上の達成が図れるという効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例のブロック図である。

15

16

【図2】図1に用いたストアインキャッシュの詳細ブロック図である。

【図3】図1に用いたディレクトリの詳細ブロック図である。

【図4】図3に用いたディレクトリメモリのフォーマット図である。

【符号の説明】

1、2 クラス

10～17 演算部

20、21 同期制御回路

30、31 共有メモリ

40、41 システムバス

100、200 ディレクトリ

101 ディレクトリメモリ

102 ディレクトリ制御回路

103 アドレス一致検出回路

104 キャンセル信号生成回路

105、106 信号線

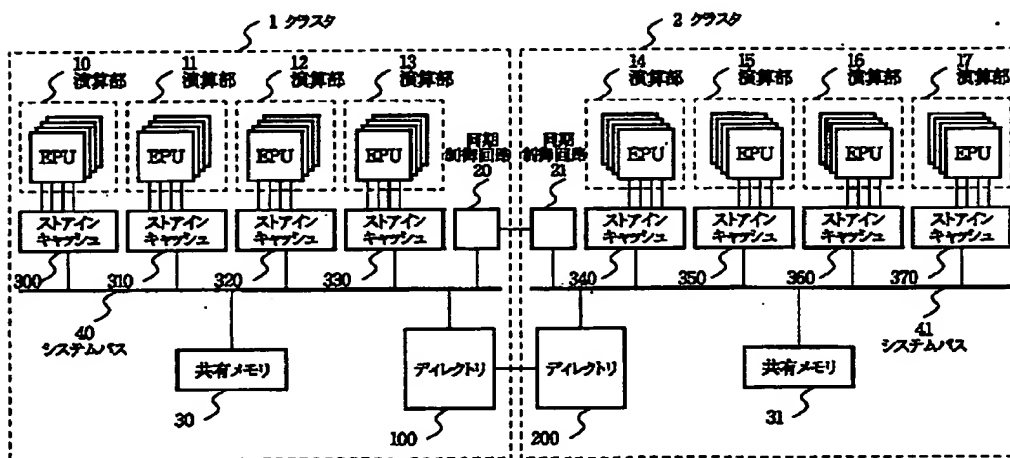
300～370 ストアインキャッシュ

301 ストアインキャッシュメモリ

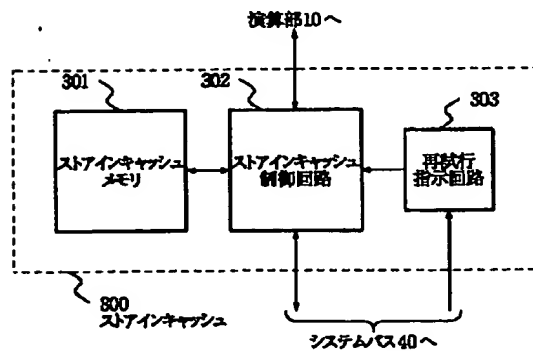
10 302 ストアインキャッシュ制御回路

303 再試行指示回路

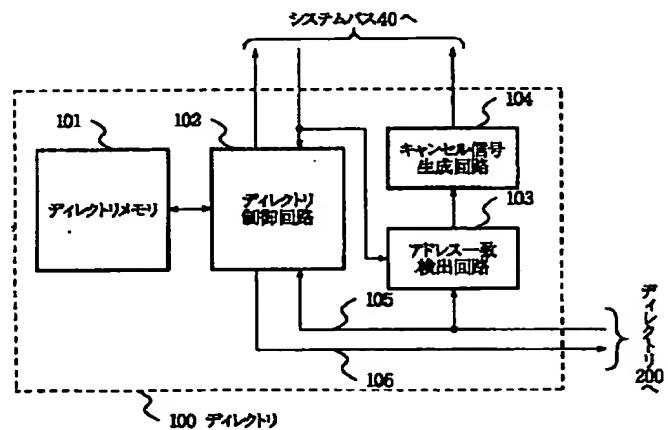
【図1】



【図2】



【図3】



【図 4】

101

0	アドレス	更新フラグ	有効フラグ
1	アドレス	更新フラグ	有効フラグ
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
N	アドレス	更新フラグ	有効フラグ